

Computer Science Unit Overview

Demographics

This unit is designed as a weekly 1 ½ hour program for third graders in a public elementary taught by an education technology specialist who is not their daily teacher.

Lesson topic

Fundamentals of computer science

Big idea

After 5 weeks, students will have a basic understanding of computer science as a discipline and feel confident creating programs in Scratch on their own.

Essential Questions

1. What is a computer?
2. How do computers “think”?
3. How can we instruct computers to perform particular tasks?
4. How can we break a task down into steps?

Knowledge

1. Students will know what computers are.
2. Students will understand the basics of how computers “think” and need to be directed.
3. Students will understand basic components of programming, such as statements (instructions), loops, and conditionals.

Skills

1. Students will be able to identify examples of computers in their lives.
2. Students will be able to break tasks down into step-by-step instructions.
3. Students will be able to detect patterns in tasks.
4. Students will be able to turn these patterns into “rules.”
5. Students will be able to turn these “rules” into instructions.

Assessment

At the very end of the unit, prompt students with the main topics of the lessons (instructions, loops, and conditionals) and see what they remember about each. Additionally, their Scratch assignments will provide insight into how much they retained and could transfer into computer-based instructions.

Lesson 1 – Intro to Computer Science

Zak Kolar
Grade 3 - Traditional classroom, 20 students
90 minutes

Objectives

1. Prepare students to learn computer science by discussing computers.
2. Demonstrate the importance of being specific when delivering instructions.

Understandings

1. Students will understand that there are many different types of computers.
2. Students will understand the basics of how computers “think.”
3. Students will be able to think about tasks in small steps.

Assessment

Collect the hand washing instructions at the end to check how specific students were with their steps. Ideally, they could be interpreted literally and result in the correct procedure for hand washing.

Instructional approach

Warm up

10 minutes

Everyone, starting with the instructor, will say their name and either ask a question about computers or share their favorite thing to do with a computer. The choice is given because some students may not have firsthand experience using computers meanwhile others may be very computer savvy. This will help introduce the topic and give the instructor a sense for where the class stands.

Introduction discussion

20 minutes

- What are some examples of computers?
 - Laptop, phone, tablet, calculator, microwave, digital clock, etc.
- What do these computers have in common?
- What is different about these computers?
- Why do we use computers?
- What is a computer?
 - Definition: A computer is a machine for storing and processing information (data). It does this by following specific instructions

Making PB&J¹

20 minutes

The teacher has the necessary supplies to make a peanut butter and jelly sandwich, but doesn't know how to make the sandwich. The students must help by instructing the teacher. The teacher picks one student to direct him or her through making the sandwich and takes all instructions extremely literally (e.g. if the first instruction is "put the peanut butter on the bread," the teacher will place the peanut butter jar on the loaf of bread inside the wrapper). After the student makes a few mistakes along the lines of that, the teacher calls on another student and continues the process until the sandwich is made (or cuts it off if it gets drawn out too long).

Discussion

20 minutes

- What was hard about giving instructions to make the sandwich?
- What were some common mistakes?
- How did you break the steps down?
- How did you know if you made a mistake in the instructions?
- How does this relate to computers?
 - Computers need specific instructions to perform action.
- What are some ways that you give instructions to a computer?
 - Tapping screen, clicking mouse, pressing buttons, etc.
- How can you give a computer many instructions at once?
 - Computer programs – many lines, each line has one specific instruction
 - Programming language – ways we can "talk" to computers, just like we use English to talk to each other.
- Computers need very specific instructions, one step at a time.
- They don't "know" what you want them to do, only what you tell them to do.
- How do you can you make sure your instructions are specific enough?

Activity

20 minutes

Students get into groups of 3 by counting off. Each group will write down instructions for a person to wash his or her hands, being as specific as possible. The instructor will circulate and assist as necessary.

Skills to be emphasized

- Thinking like a programmer – breaking large tasks into smaller steps

Materials needed

- PB&J supplies (or substitute)

Sponge activity

If there is extra time, follow groups' instructions to wash your hands in the same manner as the sandwich activity.

¹ The food might need to change based on allergies – check on this ahead of time. If there are too many allergies, this can be replaced with a non-food-related task. The important aspect is that the task requires many specific steps that may be overlooked

Lesson 2 – Giving Instructions

Zak Kolar
Grade 3 - Traditional classroom, 20 students
90 minutes

Objectives

1. Prepare students to talk about delivering instructions through a warmup activity.
2. Introduce message passing through delivering instructions.
3. Demonstrate message passing.

Understandings

1. Students will understand the basics of passing messages to manipulate objects.
2. Students will be able to plan a sequence of actions to be executed by physical objects.

Assessment

Check if the teams can successfully write a sequence of steps to navigate through the grid (see the “grid navigation” game below).

Instructional approach

Warm up

15 minutes

Play Streets and Alleys as defined by Viola Spolin in *Theater Games for the Classroom*².

Discussion

15 minutes

- In what way did the caller control the streets/alleys?
- In what way did the caller control the cat/mouse?
- What did the caller need to do to control the game?

Grid navigation³

20 minutes

- Game instructions
 - Use masking tape to create a 6' x 6' grid of boxes approximately 1' each.
 - Use tape to create Xs in some boxes.
 - Place a target object in one of the boxes.
 - One student is the mover. The mover starts in a box away from the target object.
 - One or two students are navigators. The navigators give the mover one-step directional instructions (i.e. step forward, turn left, or turn right).
 - The mover can only step to one connected box at a time (no diagonal steps) and must wait for the next instruction before each movement.
 - Using these instructions, the navigators attempt to navigate the mover to the target.
 - Repeat with several sets of navigators and movers.

² Spolin, Viola. *Theater Games for the Classroom: A Teacher's Handbook*. Evanston, IL: Northwestern UP, 1986. Print.

³ This is a modified version of the computer puzzle found here: <https://studio.code.org/s/course3/stage/2/puzzle/1> The goal of the computer version is to navigate a zombie to a sunflower by dragging directional instructions to the screen and running them

Grid navigation: team planning

25 minutes

- Modified game play
 - Break students into teams.
 - Teams get a handout (see appendix) with the layout of the grid and write a list of instructions to navigate the mover to the target.
 - Unlike the first part, students will plan the entire sequence of instructions before giving any instructions to the mover.
 - This time, the grid may have more obstacles than before since students should grasp the general concept.
 - After a team finishes writing their steps, they will send one representative to be the new navigator. The navigator still delivers each step one at a time and is not allowed to deviate from what is written on the paper.
 - If the team's instructions were unsuccessful and did not get the mover to the target, they can rewrite them and try again.
 - Teams will execute their instructions as they finish instead of presenting all at once so that the teams who need to rewrite their instructions can do so seamlessly.

Discussion

15 minutes

- Which version of the grid navigation game was easier? Which did you like better?
- What was your strategy for the first version of the game?
- What was your strategy for the second version of the game?
- Can you think of any way that would make writing the instructions easier?
 - This can be steered to the possibility of repeating steps as loops (e.g. instead of writing "step forward, step forward, step forward," write "step forward 3 times").

Skills to be emphasized

- Programming basics such as message passing
- Breaking tasks into smaller tasks
- Translating smaller tasks into instructions

Materials needed

- Masking tape
- 6' x 6' open space
- Handouts with a "map" of the grid layout for the team planning portion (see appendix)

Sponge activity

If there is extra time, play one more round of "Grid Navigation: Team Planning," with repeated steps as described above.

Lesson 3 – Loops

Zak Kolar
Grade 3 – Computer lab, 20 students
90 minutes

Objectives

1. Remind students about the lessons learned in the last class session.
2. Demonstrate the use of loops.

Understandings

1. Students will understand how loops work.
2. Students will be able to identify places in instructions to add loops.

Assessment

Check to see that students can successfully translate the instructions into handshakes (and vice versa time permitting).

Instructional approach

Review

15 minutes

Review the grid activity from the last class session. Instead of setting up the grid on the floor, the instructor will draw a 6x6 grid on a white/chalkboard complete with the start/end marks and Xs in some boxes. The students will direct him or her from the start to the end point several times. The last time, the instructor will write the steps on the board as they are taken.

Discussion

15 minutes

- What are some ways we could make the instructions easier to write? Some students may remember this question from the last class.
 - Steps can be repeated (walk forward 4 times, etc.).
 - This is called a loop.
- Explain loops
 - A loop is a cycle inside a program – a step or set of steps to be repeated a certain number of times⁴.
 - Two parts: instruction(s) and number of times to repeat

⁴ Loops can also execute indefinitely until a certain condition is met. This may be worth mentioning, but will not be fully explored in the activity.

For this activity, students will help write a “program” to count to ten.

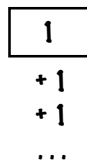
Draw a box on the board and place a “0” inside of it. Each time the students give an instruction, write the instruction under the box. Then change the value inside the box to the reflect the result of following the instruction.

Ask the students what the first step is to count to ten, starting from 0.

What is the first step? **Answer: add 1.**



Instructor writes “+ 1” below the box, and changes the value inside the square to “1”. Repeat this until the value inside the square is 10.



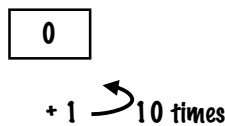
Now the instructor asks how to rewrite this as a loop.

What are the two parts of a loop? **Answer: instruction and number of times to repeat.**

What is the instruction? **Answer: add 1.**

How many times are we repeating the instruction? **Answer: 10.**

The instructor changes the value in the square back to zero. This time, he/she writes “+ 1” under and draws an arrow underneath the “+ 1”. Next to that, he/she writes “10 times.”



The instructor asks the class to keep track of how many times he/she runs through the loop. Each time, he/she updates the number in the box to reflect the current number (i.e. counts to 10). The number in the box will match the number the class says each time.

What do you need to change to make this “program” count to other numbers? **Answer: the repeat number**

The instructor and class repeat the loop with several numbers.

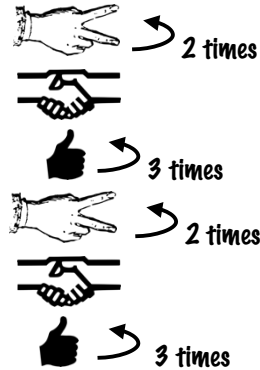
Activity⁵

15 minutes

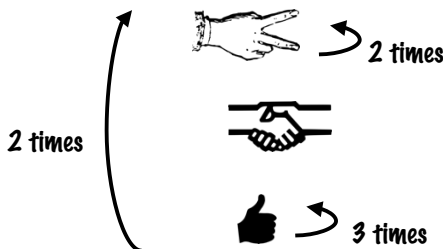
- Show students a funny long handshake video on YouTube⁶ to introduce “class handshake.”
- Perform a “class handshake” using hand motions from the hand symbol cards – see appendix:



- Ask the class for help writing “instructions” for class handshake using the symbol cards.
- Tape the cards to the board in the correct order.
- Practice the handshake with the class.
- With the class, use loops to simplify the instructions. Remove duplicates and add loops:



- Practice the handshake again to verify it hasn't changed.
- *Can this be simplified anymore?*
- Yes – the three symbols and loops repeat themselves. Loops can go inside of loops:



- Practice the handshake again to verify it hasn't changed.

Activity – partners

20 minutes

Students work with partners to create their own handshakes using the cards. They write each one down including loops, practice it, and then work on another one. Teacher circulates to help with loop notation.

Discussion

10 minutes

- How do you know when you should use a loop?
- How many loops can you put inside of other loops?
 - As many as you want, but be careful because they start to take a long time very quickly the more you put inside of each other!

⁵ Modified from “Getting Loopy” activity on Code.org: <https://code.org/curriculum/course2/5/Teacher>

⁶ https://www.youtube.com/watch?v=YNs58_61pew

Skills to be emphasized

- Identifying places to simplify code through a loop instead of repetition

Materials needed

- Chalkboard/whiteboard with writing utensil
- Hand symbols (see appendix)

Sponge activity

If there is extra time, partners can show their handshakes to other groups. The other groups write the instructions (including loops) from watching the handshake.

Lesson 4 – Conditionals

Zak Kolar
Grade 3 – Computer lab, 20 students
90 minutes

Objectives

1. Use games with which students are already familiar to illustrate “if” statements in “real life.”
2. Demonstrate the effects of different permutations of if/then/else with and/or statements.

Understandings

1. Students will understand how conditional statements can be used to filter information based on a trait.
2. Students will understand how to check multiple traits at once with different combinations of and/or.

Assessment

Make sure each student can come up with his or her own if statement during the M&M activity (and assist with any issues).

Instructional approach

Warm up

15 minutes

Play several rounds of Duck Duck Goose.

Discussion

10 minutes

- What are the rules for Duck Duck Goose?
- When the person who is “it” touches your head, how do you know what to do?
 - Listen to whether they say “duck” or “goose.”
- What do you do if they say “duck”?
 - Nothing.
- What do you do if they say “goose”?
 - Run around the circle and try to sit down before they tag you.
- How do you know who’s “it” after the two people run around the circle?
 - If “it” tags the person, the other person is the new “it”. Otherwise, “it” is “it” for another round.
- Describe if statements in programming
 - 2-3 parts: if, then, otherwise (otherwise is optional)
 - **IF** “it” says “Duck”, **THEN** you do nothing. **OTHERWISE**, you run around the circle.
 - **IF** the person running gets tagged, **THEN** they are “it”. **OTHERWISE**, they sit back down
 - Questions?

- Give each student a cup of 20 M&Ms and a napkin.
- Tell students that they WILL get to eat the M&Ms in a few minutes, but not until the activity is done.
- Students pour the M&Ms onto the napkin.
- Students each point to a random M&M and use the following rule:
 - **IF** the M&M is blue, **THEN** turn it so the M faces up. **OTHERWISE**, turn it so the M faces down.
- Repeat this process for all 20 M&Ms.
- Students mix the M&Ms back together on the napkin.
- Repeat the same process as before, using this rule:
 - **IF** the M&M is brown, **THEN** put it in the cup
 - This time, there is no otherwise. The M&Ms just stay on the napkin if they are not brown.
- Students pour the brown M&Ms back with the rest on the napkin, mixing them up in the process.
- Repeat the process again using this rule:
 - **IF** the M&M is red, **THEN** put it in the cup. **OTHERWISE:**
 - **IF** the M&M is yellow, **THEN** turn it so the M is facing down on the napkin. **OTHERWISE**, turn it so the M is facing up on the napkin.
 - Now, there are two **IFs**. Instead of saying "**OTHERWISE: IF**", we can say "**ELSE IF**."
- Students mix the M&Ms into the cup this time.
- Students close their eyes and pick one M&M out of the cup and use the following rule:
 - **IF** the M&M is green, **THEN** eat it. **OTHERWISE**, put it on the napkin.
 - Ask students to raise their hands if they ate an M&M so the class can see that not everyone got the same result
- Repeat the same instruction: Students each close their eyes and pick out one M&M from the cup and use the following rule:
 - **IF** the M&M is green, **THEN** eat it. **OTHERWISE**, put it on the napkin.
 - Ask students to raise their hands if they ate an M&M this time for everyone to see. Ask students to raise their hands if they ate an M&M the first time for everyone to see.
 - Even though the rule stayed the same, some of the people who ate M&Ms changed.
- Students use the following rule:
 - **IF** you have eaten zero M&Ms so far, **THEN** pick out two M&Ms to eat. **ELSE IF** you have eaten one M&M so far, **THEN** pick out one M&M to eat.
 - This time, there was an **ELSE IF** but no **OTHERWISE**
- Each student in the class gets a chance to be a rule maker. For each student in the class, everyone else picks out an M&M and the rule maker creates a rule using an if statement.
- Once the whole class is done, students can eat the rest of their M&Ms.

Discussion

10 minutes

- What are some different combinations of **IF**, **ELSE IF**, **THEN**, and **OTHERWISE**?
- What are some other cases where **IF** statements might be useful?
- What happens if you don't have an **OTHERWISE**?
 - The item you're looking at may not change.
- What happens if you want to check multiple things?
 - You can use **AND/OR** in your **IF**.
 - **IF** your M&M is green **OR** blue, **THEN** eat it.
 - **IF** you have an M&M on your napkin **AND** it is green, **THEN** put it back in the cup.

Activity

20 minutes

This game requires a larger space than a typical classroom such as a gym, cafeteria, or large open space. It is placed at the end of the lesson to allow time to relocate if necessary.

Play a game of "Uncle Sam:"

- One person is "Uncle Sam". Everyone but Uncle Sam lines up along one side of the room facing the middle of the room. Uncle Sam stands in the middle of the room facing everyone else.
- Everyone on the side chants, "Uncle Sam! Uncle Sam! May we cross your river dam?"
- Uncle Sam responds "Only if... " with some characteristic that pertains to certain people in the line. People who have that characteristic must attempt to run across to the other side of the room without being tagged by Uncle Sam. For example, he may respond "Only if you are wearing the color pink" and anyone wearing the color pink would need to run.
- People who get tagged by Uncle Sam follow him to the middle to help tag other people. While everyone in the middle can tag new people, only Uncle Sam decides who crosses the river dam.
- The last person left not tagged is Uncle Sam for the next game.

To help with the concept, the condition to cross the river dam must be phrased as an **IF** statement and may include an **AND/OR** (e.g. "**IF** you are wearing blue **OR** green, **THEN** you may cross my river dam").

Discussion

10 minutes

- How do and/or statements complicate the if statements?
- What is the difference between and/or statements? Which one is more restrictive?
- Any questions about if statements?

Skills to be emphasized

- Thinking in terms of Boolean statements (i.e. statements that are either true or false)
- Complex if statements

Materials needed

- Big bag of M&Ms (or Skittles, Fruit Loops, or any other food that comes in multiple colors – check with allergies. If there are too many allergies to accommodate, this can be done with non-edible, multi-colored objects such as beads)
- Cups
- Napkins

Sponge activity

If there is extra time, play Uncle Sam again, with even more intricate rules. For example, inequalities, compound and/or, etc. Give examples of these for the students so they understand what you are looking for.

- “IF you are wearing a shirt with AT LEAST two buttons, THEN you may cross my river dam.”
- “IF you are wearing a shirt AND it is green OR you are wearing sneakers AND they are yellow, THEN you may cross my river dam.”

Lesson 5 – Intro to Scratch

Zak Kolar
Grade 3 – Computer lab, 20 students
90 minutes

Objectives

1. Prepare students to tell stories through a warm up activity.
2. Demonstrate how to use the Scratch editor to modify an existing story.
3. Help students as they work on their own to tell a new story.

Understandings

1. Students will gain a more in-depth grasp of passing messages to manipulate objects through Scratch.
2. Students will be able to break down a story into a sequence of actions to execute.

Assessment

Check to what extent students can manipulate objects and characters in their story by playing the finished product (either publicly or privately, depending on time and interest). Ideally, students should incorporate techniques from previous lessons including conditionals and loop, but assess on an individual basis factoring in time and issues that have surfaced in the process.

Instructional approach

Warm up

15 minutes

Tell a story as a class⁷. The class sits in a circle and one person gives an opening sentence. Each person around the circle contributes a sentence until everyone has gone and the story is complete.

⁷ Warm up used in *Education through Creativity and Theatrical Pedagogy* class taken at Brandeis University, taught by Jen Cleary.

Introduce Scratch⁸

30 minutes

- Demonstrate Scratch by creating the class story in a new project. Depending on its complexity, the story may need to be simplified for the demonstration. If there are many characters/settings, focus on one specific setting and character for demonstration purposes.
- Emphasize the following Scratch techniques:
 - Inserting a character
 - Dragging commands to the character
 - Move
 - Speak
 - Manipulating the environment
 - Adding characters
 - Changing background
 - Navigating between settings
 - Loop blocks
 - Conditional blocks

Continuing the story: partner coding

30 minutes

- Students will work with their neighbors in pairs to continue filling in the story. Circulate around the room and help students as questions arise.

Discussion

15 minutes

The following questions will be posed to the class:

- How did you decide what to do next in the story?
- What did you do to represent this in Scratch?
- What was easy to do with Scratch?
- What was hard to do with Scratch?

Skills to be emphasized

- Programming basics such as giving commands and using loops
- Logical, sequential thinking to form an executable plan

Materials needed

- Computer for each pair of students
 - If not enough computers are available, the groups will increase to 3-4 students.
 - If there are still not enough computers for the larger groups, the teacher will continue to “drive” the computer, but students will instruct the teacher how to proceed

Sponge activity

If there is extra time, students can share their versions of the story with the class.

⁸ Scratch (<https://scratch.mit.edu/projects/editor/>) is a “block-based” coding platform designed and built by MIT. It allows students to manipulate characters and scenes by dragging blocks with commands onto the screen, and is an excellent precursor to programming.

Appendix

Lesson 2 – Grid Navigation

Instructions: On a separate piece of paper, write a set of instructions to get from the “START” box to the “END” box. You can only move one box at a time and you can’t move diagonally. You can’t leave the boxes or step on a box that has an “X” inside. You may use the following commands:

- Step forward
- Turn left
- Turn right

START			X		
	X		X	X	
	X				
			X		X
	X	X			
				X	END

Lesson 3 - Hand Symbols

